

Stochastic Patching Process

Xuhui Fan*, Bin Li†, Yi Wang, Yang Wang, Fang Chen
Data61, CSIRO

Abstract

Stochastic partition models tailor a product space into a number of rectangular regions such that the data within each region exhibit certain types of homogeneity. Due to constraints of partition strategy, existing models may cause unnecessary dissections in sparse regions when fitting data in dense regions. To alleviate this limitation, we propose a parsimonious partition model, named Stochastic Patching Process (SPP), to deal with multi-dimensional arrays. SPP adopts a “bounding” strategy to attach rectangular patches to dense regions. SPP is self-consistent such that it can be extended to infinite arrays. We apply SPP to relational modeling and use MCMC sampling for approximate inference. In particular, Conditional-SMC is adopted to sample new patches. The experimental results validate its merit compared to the state-of-the-arts.

1 Introduction

Stochastic partition processes on a product space have found many real-world applications, such as relational modeling [11, 2], community detection [20, 10], collaborative filtering [22], and random forests [12]. By tailoring the product space into rectangular regions, the partition model aims to fit data using these “blocks” such that the data within each block exhibit certain types of homogeneity. As one can choose an arbitrarily fine resolution of partition, the data can be fitted reasonably well.

The cost of data fitness is that the partition model may cause unnecessary dissections in sparse regions. Compared to regular-grid partitions, the Mondrian process (MP) [25] is a hierarchical partition process which has been more parsimonious for data fitting. However, the strategy of recursively cutting the space still cannot avoid unnecessary dissections in sparse regions. Take community detection for example, where a “block” corresponds to a community: When tailoring a block out of the relational matrix, cutting-based models will unavoidably separate some uninvolved users. As a result, some meaningless communities are generated as an undesired by-product (see Figure 1 for an illustration).

Instead of “cutting”, we propose a bounding-based partition process, named Stochastic Patching Process (SPP), to alleviate the above limitation. SPP attaches patches on a multi-dimensional array to enclose dense regions. In this way, “significant” regions of the space can be comprehensively modeled. Each patch can be generated by an outer product of multiple binary vectors, with a segment of consecutive “1” entries to indicate the initial and terminal positions of the patch. As patches are independently generated, the layout of patches can be quite flexible. This improves its expressiveness to describe those regions with complicated patterns. As a result, SPP is able to use fewer blocks (thus a more parsimonious expression of model) than those cutting-based partition models to achieve similar modeling capability.

An important property of SPP is self-consistency. This means that, by restricting the patches generated from an SPP on a multi-dimensional array Y to its sub-array X , the resulting patches are distributed as if they are directly generated on X through another SPP (given the same budget). The property will be verified in three steps: (1) the number distribution of nonempty patches is self-consistent; (2) the position distribution of a nonempty patch is self-consistent; (3) based on the

* xhfan.ml@gmail.com

† libin82cn@gmail.com

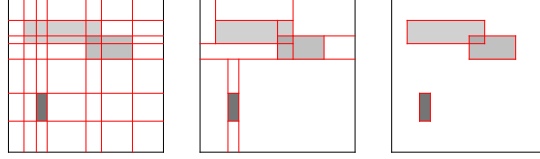


Figure 1: (left) Regular-grid partition; (middle) Hierarchical partition; (right) SPP-based partition.

above, the self-consistency of SPP can be verified. This property suggests that SPP can be extended to infinite multi-dimensional arrays according to the Kolmogorov extension theorem [5].

The merit of SPP can be seen in many applications. In this paper we investigate its merit in relational modeling, where patches can be viewed as communities while the “price” of a patch (cost per unit area) can be viewed as interacting intensity within the community. An MCMC based approximate inference method is proposed for the SPP relational model. In particular, Conditional-Sequential Monte Carlo [4] is adopted to sample completely new patches in each iteration. The experimental results on a number of real-world relational data sets demonstrate that SPP can achieve parsimonious partitions with competitive performance compared to the state-of-the-arts.

2 Preliminaries

2.1 Stochastic Partition Processes

Stochastic partition processes partition a product space into blocks. A popular application of such processes is modeling relational data such that the intensity of interactions is homogeneous within each block. In terms of partitioning strategy, state-of-the-art stochastic partition processes can be roughly categorized into regular-grid partitions and flexible axis-aligned partitions.

A regular-grid stochastic partition process is constituted by two separate partition processes on each dimension of the multi-dimensional array. The resulting orthogonal interactions between two dimensions will exhibit regular grids, which can represent interacting intensities. Typical regular-grid partition models include the infinite relational model (IRM) [11] and the infinite extension of mixed-membership stochastic blockmodels [2]. Regular-grid partition models are widely used in real-world applications for modeling graph data [9, 26].

To our knowledge, only the Mondrian process (MP) [25, 23] and the rectangular tiling process (RTP) [19] can produce flexible axis-aligned partitions on a product space. MP recursively generates axis-aligned cuts on a unit hypercube and partitions the space in a hierarchical fashion known as k -d-tree ([24] also considers a tree-consistent partition model but it is not Bayesian nonparametric). Different from the hierarchical partitioning strategy, RTP generates a flat partition structure on a two-dimensional array by assigning each entry to an existing block or a new block in sequence, without violating the rectangular restriction on the blocks.

2.2 Kolmogorov Extension Theorem

We consider a projective system of stochastic partitions: Let $\{(\Omega_X, \mathcal{B}_X)\}_{X \in \mathcal{F}(\mathbb{N}^D)}$ be a family of measurable spaces, where Ω_X is the partition space, \mathcal{B}_X is a σ -algebra on Ω_X , and $\mathcal{F}(\mathbb{N}^D)$ denotes the collection of all finite sub-arrays of the infinite D -dimensional array \mathbb{N}^D . For each $X \in \mathcal{F}(\mathbb{N}^D)$, P_{\boxplus}^X is a probability measure on \mathcal{B}_X . $\mathcal{F}(\mathbb{N}^D)$ is a partially ordered set; while $X \preceq Y \in \mathcal{F}(\mathbb{N}^D)$ the projection $\pi_{Y,X}$ restricts the partition \boxplus_Y on Y into X , by keeping \boxplus_Y ’s entries within X unchanged and removing the remaining entries. For $B_X \in \mathcal{B}_X$, the pre-image under projection is defined as $\pi_{Y,X}^{-1}B_X = \{\boxplus_Y \in \Omega_Y | \pi_{Y,X}\boxplus_Y \in B_X\}$ and the projection also satisfies $\pi_{Y,X} \circ \pi_{X,W} = \pi_{Y,W}$, $W \preceq X \preceq Y$. This family defines the projective limit measurable space $(\Omega_{\mathbb{N}^D}, \mathcal{B}_{\mathbb{N}^D})$.

Theorem 1 (Theorem 3.3.6 in [6]). *For a set of probability spaces $\{(\Omega_X, \mathcal{B}_X, P_{\boxplus}^X)\}_{X \in \mathcal{F}(\mathbb{N}^D)}$ such that projection $\pi_{Y,X} : \Omega_Y \rightarrow \Omega_X$, $X \preceq Y \in \mathcal{F}(\mathbb{N}^D)$ and $P_{\boxplus}^Y(\pi_{Y,X}^{-1}B_X) = P_{\boxplus}^X(B_X)$ holds for all*

$B_X \in \mathcal{B}_X$. Then P_{\boxplus}^X can be uniquely extended to measure $P_{\boxplus}^{\mathbb{N}^D}$ on $(\Omega_{\mathbb{N}^D}, \mathcal{B}_{\mathbb{N}^D})$ as the projective limit measurable space.

The Kolmogorov extension theorem provides us a constructive way to extend SPP to the infinite D -dimensional array \mathbb{N}^D , which will be discussed in Section 4.

2.3 Exchangeable Arrays

The Aldous–Hoover theorem [8, 3] provides the theoretical foundation to model exchangeable multi-dimensional arrays conditioned on a stochastic partition model. A random 2-dimensional array is called separately exchangeable if its distribution is invariant under separate permutations of rows and columns.

Theorem 2 (Theorem 3.2 in [21]). *A random array (R_{ij}) is separately exchangeable if and only if it can be represented as follows: There exists a random measurable function $F : [0, 1]^3 \mapsto \mathcal{X}$ such that $(R_{ij}) \stackrel{d}{=} (F(\xi_i^{\text{row}}, \eta_j^{\text{col}}, \nu_{ij}))$, where $\{\xi_i^{\text{row}}\}_i, \{\eta_j^{\text{col}}\}_j$ and $\{\nu_{ij}\}_{i,j}$ are, respectively, two sequences and an array of i.i.d. uniform random variables in $[0, 1]$.*

Relational modeling based on a stochastic partition model is a typical application of the Aldous–Hoover theorem. By defining $W(\xi_i^{\text{row}}, \eta_j^{\text{col}}) := P(F(\xi_i^{\text{row}}, \eta_j^{\text{col}}, \nu_{ij}) = 1 | F)$, every exchangeable array can be represented by a random graph function [21]. The SPP relational model introduced in Section 5 is implicitly built on this theorem.

3 Stochastic Patching Process

SPP is defined on a measurable space $(\Omega_X, \mathcal{B}_X)$, $X \in \mathcal{F}(\mathbb{N}^D)$. Each element in Ω_X denotes a partition \boxplus_X , constituted by a collection of rectangular *nonempty* patches $\{\square_k\}_k$ with corresponding costs $\{m_k\}_k$, where $k \in \mathbb{N}$ indexes the patch number in \boxplus_X . In particular, a patch is defined by an outer product $\square_k := \bigotimes_{d=1}^D u_k^{(d)}$, where $u_k^{(d)} \in \{0, 1\}^{N_X^{(d)}}$ ($N_X^{(d)}$ denotes the length of the d th dimension in X) is a position indicator vector for the d th dimension of \square_k , with the constraint that $u_k^{(d)}$ only comprises a segment of $l_k^{(d)} \in \{1, \dots, N_X^{(d)}\}$ consecutive “1” entries which starts at an initial position $s_k^{(d)} \in \{1, \dots, N_X^{(d)}\}$.

Given an array X and a budget τ , we can sample a random partition from an SPP: $\boxplus_X \sim \text{SPP}(X, \tau)$. We assume that the costs of patches are *i.i.d.* sampled from the same exponential distribution, which implies there exists a homogeneous Poisson process on the time (cost) line. The generating time of each patch is uniform in $(0, \tau]$ and the number of patches has a Poisson distribution. We represent a random partition as $\boxplus_X := \{m_k, \square_k\}_{k=1}^{K_\tau} \in \Omega_X$, which is generated as follows¹:

1. Sample the number of candidate patches $\hat{K}_\tau \sim \text{Poisson}(\tau S_X)$, where $S_X = \prod_{d=1}^D N_X^{(d)}$;
2. Sample \hat{K}_τ *i.i.d.* candidate patches. For $k' = 1, \dots, \hat{K}_\tau$, $d = 1, \dots, D$
 - (a) Sample the initial position $s_{k'}^{(d)}$ of the k' th candidate uniformly from $\{1, \dots, N_X^{(d)}\}$;
 - (b) If $s_{k'}^{(d)} = 1$, the side-length $l_{k'}^{(d)}$ increments from 0 to 1 with probability 1; otherwise $l_{k'}^{(d)}$ increments from 0 to 1 with probability $(1 - \theta)$, where $\theta \in [0, 1]$;
 - (c) If $l_{k'}^{(d)}$ has incremented from 0 to 1, generate the side-length using the distribution

$$P(l_{k'}^{(d)}) = \begin{cases} \theta^{l_{k'}^{(d)}-1}(1 - \theta), & 1 \leq l_{k'}^{(d)} < L_*; \\ \theta^{l_{k'}^{(d)}-1}, & l_{k'}^{(d)} = L_*, \end{cases}$$

$$\text{where } L_* = N_X^{(d)} - s_{k'}^{(d)} + 1;$$

¹An equivalent construction is to directly generate *nonempty* patches through thinning the Poisson process which is used for generating candidate patches – See “Alternative Construction of SPP” in Appendix.

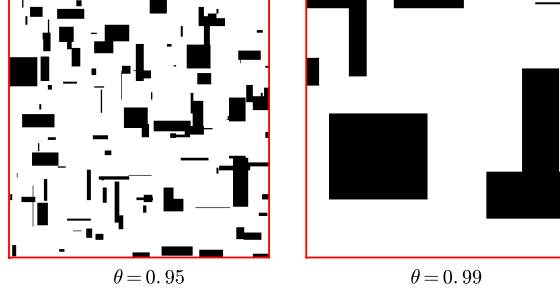


Figure 2: Two example SPP-based prior partitions on a 2-array ($N_X^{(1)} = N_X^{(2)} = 500, \tau = 0.2$). In the case of $\theta = 0.95$ (left) many small patches are generated while in the case of $\theta = 0.99$ (right) few large patches are generated.

3. Remove all empty patches and retain K_τ nonempty patches $\{\square_k | S_{\square_k} = \prod_{d=1}^D l_k^{(d)} > 0\}_{k=1}^{K_\tau}$. Sample K_τ *i.i.d.* time points uniformly in $(0, \tau]$ and index them to satisfy $t_1 < \dots < t_{K_\tau}$. Set the cost of \square_k as $m_k = t_k - t_{k-1}$ ($t_0 = 0$) and the rate² of \square_k as $\omega_k = m_k / S_{\square_k}$.

We use the initial position $s_k^{(d)}$ and the side-length $l_k^{(d)}$ of \square_k in the d th dimension to determine the position indicator vector $u_k^{(d)} \in \{0, 1\}^{N_X^{(d)}}$, which further constitutes \square_k . Thus, given K_τ , all patches are *i.i.d.* generated and the layout of patches can be quite flexible. Patches can be overlapped or even contained by others. In the two-dimensional case, a partition sampled from an SPP has the following interpretation: X can be viewed as a piece of cloth while \square_k can be viewed as a patch; the material of \square_k has rate (price) ω_k and the number of “patches” on the “cloth” is determined by S_X and the budget τ – This is where the name of “Stochastic Patching Process” comes from.

SPP has some favorable properties for being used to manipulate the prior partition via the hyperparameters. Given array X , one can use the budget τ to control the expected “volume” covered by the patches on X (overlapped parts are counted separately). Given array X and budget τ , the expected volume is a fixed constant, $\mathbb{E}(K_\tau) \cdot \mathbb{E}(S_{\square}) = \text{constant}$, and the expected number of patches $\mathbb{E}(K_\tau)$ and the expected volume of each patch $\mathbb{E}(S_{\square})$ can be balanced through θ . Formally, we have the following property (see Appendix for the proof).

Proposition 1. Let $\boxplus_X := \{m_k, \square_k\}_{k=1}^{K_\tau} \sim \text{SPP}(X, \tau)$, the expected volume of all patches is a constant in terms of the budget τ and the volume of the array X , that is, $\mathbb{E}(K_\tau) \cdot \prod_{d=1}^D \mathbb{E}(l^{(d)}) = \tau \cdot \prod_{d=1}^D N_X^{(d)}$.

Figure 2 gives an illustration of the above property based on a toy two-dimensional array, with fixed value of τ but two different values of θ . We can see that, in both cases, the volumes (black areas) covered by the patches are similar while the number of nonempty patches varies inversely to the average side-length of patch – A larger (smaller) value of θ results in a longer (shorter) expected side-length and a smaller (larger) number of nonempty patches.

Due to such a flexible layout of patches, SPP is parsimonious to model multi-dimensional arrays, especially in sparse scenarios – SPP is able to describe “significant” parts of the array (e.g. active communities in a social network) through small patches; after patching these “significant” parts, the rest are usually large and irregular sparse areas which may be neglected.

4 Self-Consistency

Section 3 has defined SPP on a finite array given a budget. To further extend SPP to the infinite array \mathbb{N}^D , an essential property of SPP is self-consistency. That is to say, while restricting an SPP on a finite D -dimensional array Y , say $\text{SPP}(Y, \tau)$, to its sub-array X , $X \subset Y \in \mathcal{F}(\mathbb{N}^D)$, the resulting

²In Section 5 we will show that such rate is useful when we use \square_k and ω_k as the priors of a community and the intensity of interactions within the community (large communities have relatively weak interactions).

patches restricted to X are distributed as if they are directly generated on X through $SPP(X, \tau)$. A typical application scenario is social network, where X and Y successively represent two snapshots of a growing network.

The self-consistency property is verified in three steps: (1) the number distribution of nonempty patches is self-consistent; (2) the position distribution of a nonempty patch is self-consistent; (3) SPP is self-consistent. Following the notations used in Sections 2.2 and 3, we use $\pi_{Y,X}$ to denote the projection that restricts $\boxplus_Y \in \Omega_Y$ to X by keeping \boxplus_Y 's entries in X unchanged and removing the rest. An “empty patch” is referred to the case $S_{\square} = 0$ ($\exists d, l^{(d)} = 0$), where S_{\square} denotes the volume of the candidate patch.

4.1 Number of Nonempty Patches

Proposition 2. *While restricting $SPP(Y, \tau)$ to X , $X \subset Y \in \mathcal{F}(\mathbb{N}^D)$, the time points of nonempty patches crossing into X from Y follows the same Poisson process for generating the time points of nonempty patches in $SPP(X, \tau)$.*

According to the definition, the candidate patches sampled from $SPP(Y, \tau)$ (or $SPP(X, \tau)$) follows a homogeneous Poisson process with intensity S_Y (or S_X). Since there exists possibility to generate empty patches, we use intensity $S_X \cdot P(S_{\square^X} > 0)$ for thinning the Poisson process to generate nonempty patches. Given the same budget τ , Proposition 2 holds if we can prove the following equality of the two Poisson process intensities

$$S_Y \cdot P(S_{\pi_{Y,X}(\square^Y)} > 0) = S_X \cdot P(S_{\square^X} > 0) \quad (1)$$

Due to the independence of dimensions, we have $P(S_{\square^X} > 0) = \prod_d P(l_X^{(d)} > 0)$. W.l.o.g, we assume that the two arrays, X and Y , have the same shape apart from the d' th dimension where Y has one additional column (the general case of more columns follows by induction). Then we can discuss two cases: X and Y 1) share the terminal boundary or 2) share the initial boundary in the d' th dimension. Eq. (1) can be proved in both cases (see Appendix for the complete proof).

Because of the same Poisson process intensity in Eq. (1), the following equality also holds

$$P_{K_{\tau}, \{m_k\}_k}^Y \left(\pi_{Y,X}^{-1} \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \right) = P_{K_{\tau}, \{m_k\}_k}^X \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \quad (2)$$

4.2 Position of Nonempty Patches

Proposition 3. *While restricting $SPP(Y, \tau)$ to X , $X \subset Y \in \mathcal{F}(\mathbb{N}^D)$, the marginal probability of the pre-images of a nonempty patch \square^X in Y (given the patches in Y crossing into X that are nonempty) equals to the probability of \square^X directly sampled from $SPP(X, \tau)$ (given the patches in X that are nonempty), that is $P_{\square}^Y(\pi_{Y,X}^{-1}(\square^X) \mid S_{\pi_{Y,X}(\square^Y)} > 0) = P_{\square}^X(\square^X \mid S_{\square^X} > 0)$.*

W.l.o.g, we assume that the two arrays, X and Y , have the same shape apart from the d' th dimension where Y has one additional column (the general case follows by induction). For dimensions $d \neq d'$, it is obvious that the law of patches is consistent under projection because the projection is the identity. Given the same budget τ , Proposition 3 holds if we can prove the following equality

$$P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) = P_u^X(u_X^{(d')}) \mid |u_X^{(d')}| \geq 1 \quad (3)$$

where $u_X^{(d')}$ indicates the initial position, $s_X^{(d')}$, and the side-length, $l_X^{(d')}$, of the d' th side of \square^X ; $|u_X^{(d')}| \geq 1$ means that there is at least one “1” entry in $u_X^{(d')}$.

Eq. (3) involves four cases in total: X and Y 1) share the initial boundary or 2) share the terminal boundary in the d' th dimension; in each case, there are two sub-cases regarding whether the terminal position (for Case 1) or the initial position (for Case 2) of the d' th side of \square^X locates at the boundary of X . We can prove that Eq. (3) holds in all cases. Consider all D dimensions we have $P_{\square}^Y(\pi_{Y,X}^{-1}(\square^X) \mid S_{\pi_{Y,X}(\square^Y)} > 0) = P_{\square}^X(\square^X \mid S_{\square^X} > 0)$ (see Appendix for the complete proof).

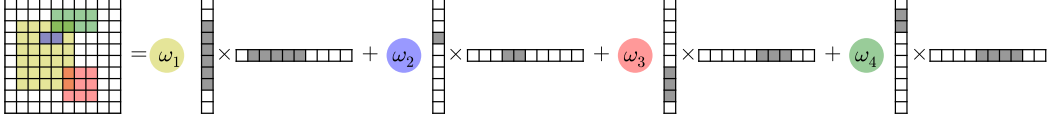


Figure 3: SPP relational model: $\{\omega_k\}_k$ denote intensities of relations within communities $\{\square_k\}_k$.

4.3 Self-Consistency of SPP

Now we are ready to prove $P_{\boxplus}^Y(\pi_{Y,X}^{-1}(\boxplus_X)) = P_{\boxplus}^X(\boxplus_X)$.

$$\begin{aligned}
& P_{\boxplus}^Y(\pi_{Y,X}^{-1}(\boxplus_X)) \\
&= P_{K_\tau, \{m_k\}_k}^Y \left(\pi_{Y,X}^{-1} \left(K_\tau^X, \{m_k^X\}_{k=1}^{K_\tau^X} \right) \right) \cdot \prod_{k=1}^{K_\tau^X} P_{\square}^Y \left(\pi_{Y,X}^{-1}(\square_k^X) \mid S_{\pi_{Y,X}(\square_k^Y)} > 0 \right) \\
&\stackrel{(*)}{=} P_{K_\tau, \{m_k\}_k}^X \left(K_\tau^X, \{m_k^X\}_{k=1}^{K_\tau^X} \right) \cdot \prod_{k=1}^{K_\tau^X} P_{\square}^X(\square_k^X \mid S_{\square_k^X} > 0) \\
&= P_{\boxplus}^X(\boxplus_X)
\end{aligned}$$

where $P_{K_\tau, \{m_k\}_k}^X(\cdot)$ and $\prod_{k=1}^{K_\tau^X} P_{\square}^X(\cdot)$ in the step marked by $(*)$ are obtained by applying Propositions 2 and 3, respectively. According to the Kolmogorov extension theorem (see Section 2.2), we have the following result.

Theorem 3. *The probability measure P_{\boxplus}^X on a measurable space $(\Omega_X, \mathcal{B}_X)$ of SPP, $X \in \mathcal{F}(\mathbb{N}^D)$, can be uniquely extended to $P_{\boxplus}^{\mathbb{N}^D}$ on $(\Omega_{\mathbb{N}^D}, \mathcal{B}_{\mathbb{N}^D})$ as the projective limit measurable space.*

5 Application to Relational Modeling

5.1 SPP Relational Model

A typical application of SPP is relational modeling. Given the relational data as an asymmetric matrix $R \in \{0, 1\}^{N \times N}$, with R_{ij} indicating the relation from node i to node j , the patches $\{\square_k\}_k$ with different rates $\{\omega_k\}_k$ of a partition \boxplus are used for modeling communities with different intensities of relations. Because $\{\square_k\}_k$ can be overlapped, the intensity of relations in an overlapped part on R is synthesized by the rates of the involved patches (see Figure 3).

The generative process of an SPP relational model is as follows: (1) Generate a partition \boxplus ; (2) For $i = 1, \dots, N$, generate row index r_i of R ; (3) for $j = 1, \dots, N$, generate column index c_j of R ; (4) for $i, j = 1, \dots, N$, generate relational data $R_{r_i c_j} \sim \text{Bernoulli}(\sigma(\sum_{k=1}^{K_\tau} \frac{\omega_k}{\gamma} \cdot u_{k,i}^{(1)} u_{k,j}^{(2)}))$, where $\sigma(x) = \frac{\exp(x+e^{-6})-1}{\exp(x+e^{-6})+1}$ is a selected function for mapping the aggregated rate from $[0, \infty)$ to $(0, 1)$ as intensity of relations and γ is a scaling parameter. While here we instantiate an SPP relational model with binary interactions (Bernoulli likelihood), other types of relations (e.g., Categorical likelihood) can also be plugged in.

Actually, SPP and the mapping function $\sigma(\cdot)$ play together as the role of random function $W(\cdot)$ defined in Section 2.3. The uniformly exchanged row and column indices (r_i and c_j) resemble the row and column indices (ξ_i^{row} and η_j^{col}) which are uniformly sampled in $[0, 1]$. By re-arranging the rows and columns of R according to the inferred indices, the SPP relational model is expected to uncover homogeneous interactions in R as compact patches.

5.2 Sampling for SPP Relational Model

Algorithm 1 Sampling for SPP Relational Model

Input: Relational data R , budget τ , hyper-parameters θ, γ , iteration time T

Output: $K_\tau, \{m_k, u_k^{(1)}, u_k^{(2)}\}_k, \{r_i\}_i$ and $\{c_j\}_j$

```
for  $t = 1, \dots, T$  do
  Sample  $K_\tau$ ;
  Sample  $\{m_k\}_{k=1}^{K_\tau}$ ; // Metropolis-Hastings
  for  $k = 1, \dots, K_\tau$  do
    Sample  $(u_k^{(1)}, u_k^{(2)})$ ; // C-SMC (Algorithm 2)
  end for
  Sample  $\{r_i\}_{i=1}^N, \{c_j\}_{j=1}^N$ ; // Multiple-Try Metropolis
end for
```

The joint probability of the data $\{R_{ij}\}_{i,j}$, the number of nonempty patches K_τ , the variables of the nonempty patches $\{m_k, u_k^{(1)}, u_k^{(2)}\}_{k=1}^{K_\tau}$, and the indices $\{r_i\}_i, \{c_j\}_j$ gives

$$\begin{aligned} & P(\{R_{ij}\}_{i,j}, K_\tau, \{m_k, u_k^{(1)}, u_k^{(2)}\}_k, \{r_i\}_i, \{c_j\}_j | \theta, \tau, \gamma, N) \\ &= \prod_{i,j} P(R_{ij} | K_\tau, \{m_k, u_k^{(1)}, u_k^{(2)}\}_k, \gamma) \\ &\quad \cdot P(\{r_i\}_i | N) \cdot P(\{c_j\}_j | N) \cdot P(K_\tau, \{m_k\}_k | \tau, \theta, N) \\ &\quad \cdot \prod_k P(u_k^{(1)} | \theta, N) P(u_k^{(2)} | \theta, N) \end{aligned}$$

where $P(R_{ij} | K_\tau, \{m_k, u_k^{(1)}, u_k^{(2)}\}_k, \gamma) = \rho_{ij}^{R_{ij} c_j} (1 - \rho_{ij})^{1 - R_{ij} c_j} = \ell(r_i, c_j, \rho_{ij})$ refers to the probability of $R_{ij} c_j$ and $\rho_{ij} = \sigma(\sum_{k=1}^{K_\tau} \frac{\omega_k}{\gamma} \cdot u_{k,i}^{(1)} u_{k,j}^{(2)})$ denotes the parameter of the Bernoulli likelihood; $P(\{r_i\}_i | N) = P(\{c_j\}_j | N) = \frac{1}{N!}$ denotes the probability of row and column indices; $P(K_\tau, \{m_k\}_k | \tau, \theta, N) = (\gamma N^2 \theta_*)^{K_\tau} e^{-\tau \gamma N^2 \theta_*}$ (where $\theta_* = \frac{1}{N^2} \cdot [\theta + N(1 - \theta)]^2$) denotes the joint probability of the number and the time points of the nonempty patches.

We adopt MCMC methods for sampling the posteriors of $K_\tau, \{m_k, u_k^{(1)}, u_k^{(2)}\}_k, \{r_i\}_i$ and $\{c_j\}_j$. By an abuse of notation, in the following $\rho_{ij}^{x \rightarrow x^*}$ is used to represent the case that the likelihood is updated by replacing x with x^* , keeping the other variables unchanged. Also, we use ρ_{ij}^{-k} to denote the likelihood computed excluding the k th patch. The sampling algorithm is outlined in Algorithm 1.

Sample K_τ We use a similar strategy of [1] for updating K_τ . First, we use probability $P_0 = \frac{1}{2}$ (or $1 - P_0$) to choose proposing adding (or removing) a nonempty patch. The proposal probability of adding a nonempty patch is

$$q_{\text{add}}(K_\tau \rightarrow K_\tau + 1) = P_0 \cdot \frac{1}{\tau} \cdot P(\square_*) \quad (4)$$

where \square_* denotes a newly added patch; the proposal probability of deleting an existing patch is

$$q_{\text{del}}(K_\tau \rightarrow K_\tau - 1) = \frac{1 - P_0}{K_\tau}. \quad (5)$$

We accept adding or removing a patch with a ratio of $\min(1, \alpha_{\text{add}})$ or $\min(1, \alpha_{\text{del}})$, where

$$\alpha_{\text{add}} = \frac{\prod_{i,j} \ell(r_i, c_j, \rho_{ij}^{K_\tau \rightarrow K_\tau + 1})}{\prod_{i,j} \ell(r_i, c_j, \rho_{ij})} \cdot \frac{\tau \gamma N^2 \theta_*}{K_\tau + 1} \cdot \frac{1 - P_0}{P_0} \quad (6)$$

$$\alpha_{\text{del}} = \frac{\prod_{i,j} \ell(r_i, c_j, \rho_{ij}^{K_\tau \rightarrow K_\tau - 1})}{\prod_{i,j} \ell(r_i, c_j, \rho_{ij})} \cdot \frac{K_\tau}{\tau \gamma N^2 \theta_*} \cdot \frac{P_0}{1 - P_0} \quad (7)$$

It is worth noting that in the proposal of adding a new nonempty patch, it is generated by following Step 2 (a) and (b) of “Alternative Construction of SPP” in Appendix.

Algorithm 2 C-SMC Sampler for \square_k^*

Input: Relational data R , hyper-parameters θ , current patches $\{\square_k\}_k$, number of particles C , maximum length of the sequence in C-SMC I

Output: New position of patch $\square_k^* = (u_k^{(1)}, v_k^{(2)})$

```
for  $c = 2, \dots, C$  do
    Sample random initial positions for  $\mathcal{P}_c(0)$ ;
end for
for  $i = 1, \dots, I$  do
    Set  $\mathcal{P}_1(i) = \square_k(i)$  and  $j_1 = 1$ ;
    for  $c = 2, \dots, C$  do
        Sample  $\mathcal{P}_c(i)$  from  $q(\cdot | \mathcal{P}_c(i-1))$ ;
    end for
    for  $c = 1, \dots, C$  do
        Update weight  $\omega_c(i)$  according to Eq. (9);
    end for
    Normalize  $\{\omega_c(i)\}_{c=1}^C$  and obtain  $\{\bar{\omega}_c(i)\}_{c=1}^C$ ;
    Resample indices  $\{j_c\}_{c=2}^C$  from  $\sum_{c'=1}^C \bar{\omega}_{c'}(i) \delta_{c'}$ ;
     $\forall c$ , Assign  $\mathcal{P}_c(i) = \mathcal{P}_{j_c}(i)$ ;
end for
Sample  $c^*$  from  $\sum_{c'=1}^C \bar{\omega}_{c'}(I) \delta_{c'}$  and let  $\square_k^* = \mathcal{P}_{c^*}(I)$ ;
```

Sample $\{m_k\}_k$ For the k th patch, $k \in \{1, \dots, K_T\}$, a new m_k^* is sampled from the proposal distribution, which is a truncated Exponential distribution $f(m_k^*) \propto e^{-\gamma N^2 \theta_* m_k^*} \mathbb{1}[m_k^* \in (0, \tau - \sum_{k' \neq k} m_{k'})]$. We then accept m_k^* with a ratio of $\min(1, \alpha)$, where

$$\alpha = \frac{\prod_{i,j} \ell(r_i, c_j, \rho_{ij}^{m_k \rightarrow m_k^*})}{\prod_{i,j} \ell(r_i, c_j, \rho_{ij})} \cdot \frac{e^{-\gamma N^2 \theta_* m_k}}{e^{-\gamma N^2 \theta_* m_k^*}} \quad (8)$$

Sample $\{u_k^{(1)}, u_k^{(2)}\}_k$ A straightforward way to update $\{u_k^{(1)}, u_k^{(2)}\}_k$ is to use the Metropolis-Hastings (MH) algorithm. Since MH only proposes local changes to update the current positions of $\{\square_k\}_k$ in each iteration, the sampler can explore limited space in Ω_X given a reasonable sampling budget. To overcome this problem, we adopt Conditional-Sequential Monte Carlo [4] (C-SMC) for sampling completely new positions of $\{\square_k\}_k$ in each iteration.

The description of the C-SMC sampler for \square_k is shown in Algorithm 2. C-SMC works similarly as SMC to propose high dimensional variables ($u_k^{(1)}, u_k^{(2)}$ in our case) at each stage of the sequence, except for clamping the first particle as the current patch $\mathcal{P}_1(i) = \square_k(i)$ (where $\square_k(i)$ represents the patch updated at the i th stage). The advantage of C-SMC for our problem is that it is able to propose a completely new candidate patch to replace the current one with high acceptance ratio. The key connection between C-SMC and patch sampling is to view the generative process of a patch (introduced in Section 3) as a sequence of state variables (similar as [13] using C-SMC for sampling a tree). Starting from a randomly chosen initial position, the patch undergoes entry-wise growing in both directions of row and column. Let $q(\cdot | \mathcal{P}_c(i-1))$ denote the proposal distribution of the c th particle at the i th stage, $q(\cdot | \mathcal{P}_c(i-1))$ is set as the conditional distribution of $\mathcal{P}_c(i)$ given $\mathcal{P}_c(i-1)$ under the generative prior, in both directions of row and column. As a result, the weight $\omega_c(i)$ for the c th particle at the i th stage is updated as

$$\omega_c(i) = \frac{\prod_{i',j'} \ell(r_{i'}, c_{j'}, \rho_{i'j'}^{\mathcal{P}_c(i-1) \rightarrow \mathcal{P}_c(i)})}{\prod_{i',j'} \ell(r_{i'}, c_{j'}, \rho_{i'j'})} \quad (9)$$

where i' and j' refer to the row and column indices whose likelihoods are influenced by the particle updating.

Although the complexity of the C-SMC sampler (Algorithm 2) is higher than the MH algorithm, C-SMC can propose completely new patches with high acceptance ratio. This ability enables it to fast explore the partition space Ω_X . In this sense, C-SMC can provide a much better approximation to the posterior distribution of the patches.

Sample $\{r_i\}_i, \{c_j\}_j$ To cooperate with the C-SMC sampler for higher acceptance ratio, we adopt the Multiple-Try Metropolis method [16] for sampling the row and column indices of the relational data (please refer to Appendix for details).

5.3 Experiments

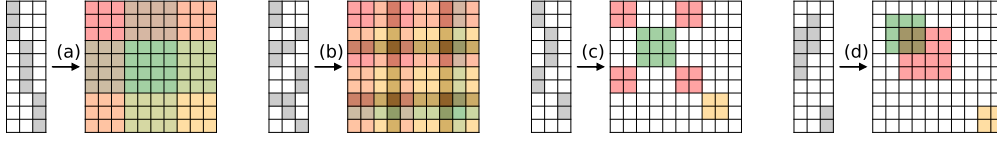


Figure 4: Partition illustration with three toy binary latent feature vectors for each method: (a) IRM presents regular grids; (b) LFRM presents plaid patterns; (c) MTA-RM presents non-overlapped noncontiguous tiles; (d) SPP-RM presents overlapped contiguous patches.

Table 1: Relational modeling (link prediction) comparison results (AUC \pm std)

| Data Sets | IRM | LFRM | MP-RM | MTA-RM | SPP-RM |
|-----------|-------------------|-------------------|-------------------|-------------------|--------------------------|
| Digg | 0.792 \pm 0.011 | 0.801 \pm 0.031 | 0.784 \pm 0.020 | 0.793 \pm 0.005 | 0.815 \pm 0.011 |
| Flickr | 0.870 \pm 0.003 | 0.881 \pm 0.006 | 0.868 \pm 0.011 | 0.872 \pm 0.004 | 0.885 \pm 0.008 |
| Gplus | 0.857 \pm 0.002 | 0.860 \pm 0.008 | 0.855 \pm 0.007 | 0.857 \pm 0.002 | 0.868 \pm 0.002 |
| Facebook | 0.872 \pm 0.013 | 0.881 \pm 0.023 | 0.876 \pm 0.028 | 0.885 \pm 0.010 | 0.889 \pm 0.018 |
| Twitter | 0.860 \pm 0.003 | 0.868 \pm 0.021 | 0.815 \pm 0.055 | 0.870 \pm 0.006 | 0.870 \pm 0.016 |

We empirically test the SPP relational model (SPP-RM) for link prediction. We compare SPP-RM with four state-of-the-arts: (1) IRM [11] (regular grids); (2) LFRM [18] (plaid grids); (3) MP Relational Model (MP-RM) [25] (hierarchical k d-tree); (4) Matrix Tile Analysis Relational Model (MTA-RM) [7] (noncontiguous tiles). All these models except MP-RM can be represented as a (weighted) sum of outer products of binary latent feature (community) vectors (see Figure 4). For IRM and LFRM, we adopt the collapsed Gibbs sampling algorithms for inference; for MP-RM, we adopt the reversible-jump MCMC algorithm for inference [27]; for MTA-RM, we adopt the Iterative Conditional Modes algorithm used in [7].

Data Sets: Five social network data sets are used: Digg, Flickr [28], Gplus [17], and Facebook, Twitter [15]. We extract a subset of nodes (top 1000 active nodes based on their interactions with others) from each data set for constructing the relational data matrix.

Experimental Setting: We set the hyper-parameters for each method as follows: In IRM, we let α be sampled from a gamma prior $\Gamma(1, 1)$ and the row and column partitions be sampled from two independent Dirichlet processes; In LFRM, we let α be sampled from a gamma prior $\Gamma(2, 1)$. As the budget parameter of MP-RM is hard to sample [14], we set it to 3, which suggests that around $(3 + 1) \times (3 + 1)$ blocks would be generated. For parametric model MTA-RM, we simply set the number of tiles to 16; In SPP-RM, we set $\theta = 0.99$ and $\tau = 0.5, \gamma = 10^{-2}$, which leads to an expectation of 12.5 patches. We use 5 particles in Algorithm 2 and set the maximum length of the sequence to 500 (half number of rows/columns). The reported performance is averaged over 10 randomly selected hold-out test sets (Train : Test = 9 : 1).

Results: Table 1 reports the performance comparison results on the five data sets. We can see that SPP-RM consistently outperforms the other four methods in all cases, with around 0.01 improvement compared to the runner-up in prediction AUC. The overall results validate that SPP-RM is effective in relational modeling due to its flexibility via attaching patches to dense regions.

Figure 5 (rows 1~5) illustrates the visual patterns of the partition results. As expected, our bounding-based method SPP-RM indeed focuses on describing dense regions of relational data matrices with fewer patches, while the two representative cutting-based methods, IRM and MP-RM, cut sparse regions into more blocks. An interesting observation of SPP-RM is that overlapped patches are very useful in describing inter-community interactions (e.g., patches in Digg, Flickr, and Gplus) and community-in-community interactions (e.g., upper-right corner in Flickr and Gplus). Thus, in addition to improved performance, SPP-RM also produce parsimonious partitions.

Figure 5 (rows 6~7) plot the average performance versus the wall-clock time for investigating the convergence behavior of the compared methods. IRM and LFRM converge fastest because of efficient collapsed Gibbs sampling. MTA-RM also converges fast because it is trained using a simple iterative algorithm. Although SPP-RM takes longer time for each sampling epoch compared to the other methods, it usually can obtain competitive performance in few iterations, thanks to the C-SMC sampler.

6 Conclusion

A parsimonious partition process, named Stochastic Patching Process (SPP), is proposed. Instead of the cutting-based strategy, we adopt a bounding-based strategy to attach *i.i.d.* rectangular patches to model dense data regions in the space such that it can avoid unnecessary dissections in sparse regions. We apply SPP to relational modeling and find that SPP can achieve clear performance gain with fewer patches (blocks) compared to the state-of-the-art relational modeling methods.

7 Acknowledgement

We thank the anonymous meta-reviewer of ICML-16 for his/her constructive and very detail comments, which helped us to significantly improve the manuscript.

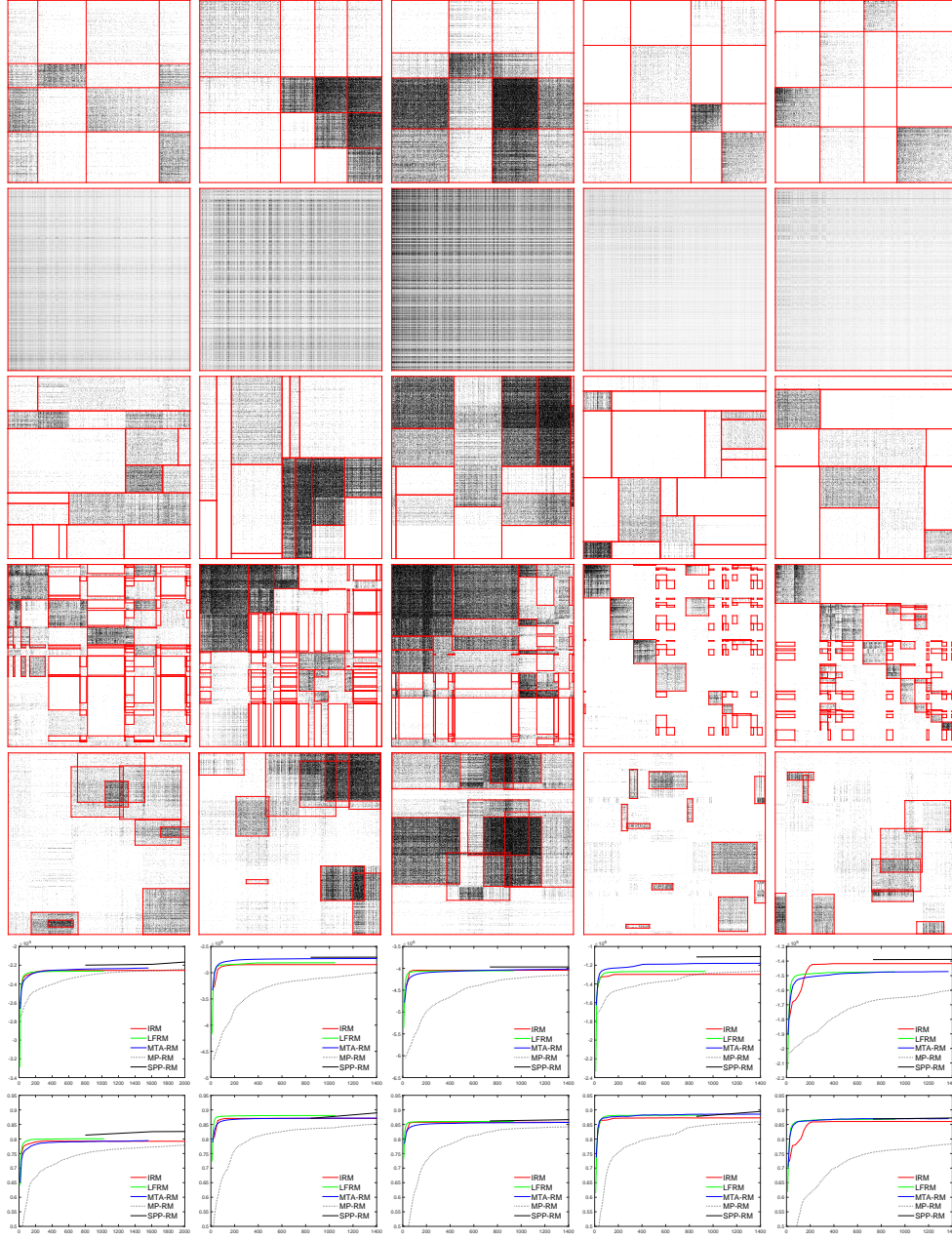


Figure 5: Partition structure visualization and performance comparison on the five data sets: (from left to right) Digg, Flickr, Gplus, Facebook and Twitter. The rows correspond to (from top to bottom) (1) IRM, (2) LFRM, (3) MP-RM, (4) MTA-RM, (5) SPP-RM, (6) training log-likelihood vs. wall-clock time (s) and (7) testing AUC vs. wall-clock time (s). Note that the curves of SPP-RM start from around 800s because the C-SMC sampling in each iteration takes longer time.

References

- [1] Ryan P. Adams, Iain Murray, and David J.C. MacKay. Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *ICML*, pages 9–16, 2009.
- [2] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. In *NIPS*, pages 33–40, 2009.
- [3] David J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.
- [4] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [5] Salomon Bochner. *Harmonic Analysis and the Theory of Probability*. University of California Press, 1955.
- [6] Kai Lai Chung. *A Course in Probability Theory*. Academic Press, 2001.
- [7] Inmar Givoni, Vincent Cheung, and Brendan Frey. Matrix tile analysis. In *UAI*, pages 200–207, 2006.
- [8] Douglas N. Hoover. Relations on probability spaces and arrays of random variables. *Preprint, Institute for Advanced Study, School of Mathematics, Princeton, NJ*, 1979.
- [9] Katsuhiko Ishiguro, Tomoharu Iwata, Naonori Ueda, and Joshua B. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *NIPS*, pages 919–927, 2010.
- [10] Brian Karrer and Mark E.J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.
- [11] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, pages 381–388, 2006.
- [12] Balaji Lakshminarayanan, Daniel M. Roy, and Yee Whye Teh. Mondrian forests: Efficient online random forests. In *NIPS*, pages 3140–3148, 2014.
- [13] Balaji Lakshminarayanan, Daniel M. Roy, and Yee Whye Teh. Particle Gibbs for Bayesian additive regression trees. In *AISTATS*, pages 553–561, 2015.
- [14] Balaji Lakshminarayanan, Daniel M. Roy, and Yee Whye Teh. Mondrian forests for large-scale regression when uncertainty matters. In *AISTATS*, pages 1478–1487, 2016.
- [15] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.
- [16] Jun S. Liu, Faming Liang, and Wing Hung Wong. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.
- [17] Julian J. McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *NIPS*, volume 2012, pages 548–556, 2012.
- [18] Kurt Miller, Michael I. Jordan, and Thomas L. Griffiths. Nonparametric latent feature models for link prediction. In *NIPS*, pages 1276–1284, 2009.
- [19] Masahiro Nakano, Katsuhiko Ishiguro, Akisato Kimura, Takeshi Yamada, and Naonori Ueda. Rectangular tiling process. In *ICML*, pages 361–369, 2014.
- [20] Krzysztof Nowicki and Tom A.B. Snijders. Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.
- [21] Peter Orbanz and Daniel M. Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(02):437–461, 2015.
- [22] Ian Porteous, Evgeniy Bart, and Max Welling. Multi-HDP: A non parametric Bayesian model for tensor factorization. In *AAAI*, pages 1487–1490, 2008.
- [23] Daniel M. Roy. *Computability, Inference and Modeling in Probabilistic Programming*. PhD thesis, MIT, 2011.
- [24] Daniel M. Roy, Charles Kemp, Vikash Mansinghka, and Joshua B. Tenenbaum. Learning annotated hierarchies from relational data. In *NIPS*, pages 1185–1192, 2007.
- [25] Daniel M. Roy and Yee Whye Teh. The Mondrian process. In *NIPS*, pages 1377–1384, 2009.
- [26] Mikkel N. Schmidt and Morten Mørup. Nonparametric Bayesian modeling of complex networks: An introduction. *IEEE Signal Processing Magazine*, 30(3):110–128, 2013.
- [27] Pu Wang, Kathryn B. Laskey, Carlotta Domeniconi, and Michael I. Jordan. Nonparametric Bayesian co-clustering ensembles. In *SDM*, pages 331–342, 2011.
- [28] Reza Zafarani and Huan Liu. Social computing data repository at ASU, 2009.

Alternative Construction of SPP

An alternative construction of SPP which is equivalent to the one introduced in Section 3 is as follows:

1. Sample the number of *nonempty* patches $K_\tau \sim \text{Poisson}(\tau S_X \cdot P(S_{\square^x} > 0))$, where $P(S_{\square^x} > 0) = \prod_{d=1}^D \frac{1}{N_X^{(d)}} \cdot [\theta + (1 - \theta)N_X^{(d)}]$;
2. Given K_τ , sample *i.i.d.* *nonempty* patches $\{\square_k\}_{k=1}^{K_\tau}$. For $k = 1, \dots, K_\tau, d = 1, \dots, D$
 - (a) Sample the initial position $s_k^{(d)}$ of \square_k from $\{1, 2, \dots, N_X^{(d)}\}$ in proportion to $\{1, (1 - \theta), \dots, (1 - \theta)\}$;
 - (b) Sample the side-length $l_k^{(d)}$ using the distribution

$$P(l_k^{(d)}) = \begin{cases} \theta l_k^{(d)-1} (1 - \theta), & 1 \leq l_k^{(d)} < L_*; \\ \theta l_k^{(d)-1}, & l_k^{(d)} = L_*, \end{cases}$$

$$\text{where } L_* = N_X^{(d)} - s_k^{(d)} + 1;$$

3. Sample K_τ *i.i.d.* time points uniformly in $(0, \tau]$ and index them to satisfy $t_1 < \dots < t_{K_\tau}$. Set the cost of \square_k as $m_k = t_k - t_{k-1}$ ($t_0 = 0$) and the rate of \square_k as $\omega_k = m_k / S_{\square_k}$, where $S_{\square_k} = \prod_{d=1}^D l_k^{(d)}$.

In this way, one can directly sample *nonempty* patches through thinning the Poisson process which is used for generating candidate patches.

Proof for Proposition 1

Proof. $\forall d \in \{1, \dots, D\}$, we have the probability of side-length $l^{(d)} > 0$ as

$$P(l^{(d)} > 0) = \frac{1}{N_X^{(d)}} + \frac{1 - \theta}{N_X^{(d)}} \cdot (N_X^{(d)} - 1) = \frac{1}{N_X^{(d)}} \cdot [\theta + (1 - \theta)N_X^{(d)}]. \quad (10)$$

According to the thinning property of the Poisson process, we thus have the following expected number of nonempty patches:

$$\mathbb{E}(K_\tau) = \tau \cdot S_X \cdot \prod_{d=1}^D P(l^{(d)} > 0) = \tau \cdot \prod_{d=1}^D [\theta + (1 - \theta)N_X^{(d)}]. \quad (11)$$

For the expectation of side-length $l^{(d)}$, we need to consider its all possible initial positions $s^{(d)}$:

$$\begin{aligned} \mathbb{E}(l^{(d)}) &= \sum_{l^{(d)}=1}^{N_X^{(d)}} l^{(d)} \cdot P_l(l^{(d)}) = \sum_{l^{(d)}=1}^{N_X^{(d)}} l^{(d)} \cdot \sum_{s^{(d)}=1}^{N_X^{(d)}-l^{(d)}+1} P(l^{(d)}|s^{(d)}) P_s(s^{(d)}) \\ &= \sum_{s^{(d)}=1}^{N_X^{(d)}} P_s(s^{(d)}) \cdot \sum_{l^{(d)}=1}^{N_X^{(d)}-s^{(d)}+1} l^{(d)} \cdot P(l^{(d)}|s^{(d)}). \end{aligned} \quad (12)$$

For convenience of notation, we let

$$E_n = P_s(s^{(d)} = n) \cdot \sum_{l^{(d)}=1}^{N_X^{(d)}-s^{(d)}+1} l^{(d)} \cdot P(l^{(d)}|s^{(d)} = n). \quad (13)$$

In the case of $s^{(d)} = 1$ (the initial position is on the boundary of X), we have $P_s(s^{(d)} = 1) = \frac{1}{1+(N_X^{(d)}-1)(1-\theta)}$ according to Step 2(a) in “Alternative Construction of SPP”; and the expected side-length is

$$E_1 = \frac{1}{1+(N_X^{(d)}-1)(1-\theta)} \left[(1-\theta) + 2\theta(1-\theta) + 3\theta^2(1-\theta) + \dots \right. \\ \left. + (N_X^{(d)}-1)\theta^{N_X^{(d)}-2}(1-\theta) + N_X^{(d)}\theta^{N_X^{(d)}-1} \right]. \quad (14)$$

To simplify the above equation, we can compute $E_1 - \theta E_1$ to cancel out many terms and obtain the following equation

$$E_1 - \theta E_1 = \frac{1-\theta}{\theta + (1-\theta)N_X^{(d)}} \left[1 + \theta + \theta^2 + \dots + \theta^{N_X^{(d)}-1} \right]. \quad (15)$$

Thus we have

$$E_1 = \frac{1}{\theta + (1-\theta)N_X^{(d)}} \cdot \frac{1 - \theta^{N_X^{(d)}}}{1 - \theta}. \quad (16)$$

Similarly, in the case of $s^{(d)} = n \in \{2, 3, \dots, N\}$ (the initial position is not on the boundary of X), we have

$$E_n = \frac{1-\theta}{\theta + (1-\theta)N_X^{(d)}} \cdot \frac{1 - \theta^{N_X^{(d)}-n+1}}{1 - \theta}. \quad (17)$$

Now the expectation of $l^{(d)}$ becomes

$$\mathbb{E}(l^{(d)}) = \frac{1}{\theta + (1-\theta)N_X^{(d)}} \cdot \frac{1 - \theta^{N_X^{(d)}}}{1 - \theta} + \frac{1-\theta}{\theta + (1-\theta)N_X^{(d)}} \cdot \sum_{n=2}^{N_X^{(d)}} \frac{1 - \theta^{N_X^{(d)}-n+1}}{1 - \theta} \\ = \frac{1}{\theta + (1-\theta)N_X^{(d)}} \left[\frac{1 - \theta^{N_X^{(d)}}}{1 - \theta} + \sum_{n=1}^{N_X^{(d)}-1} (1 - \theta^n) \right] \\ = \frac{N_X^{(d)}}{\theta + (1-\theta)N_X^{(d)}}. \quad (18)$$

By combining Eq. (11) and Eq. (18), we can obtain the expected volume of all patches:

$$\mathbb{E}(K_\tau) \cdot \prod_{d=1}^D \mathbb{E}(l^{(d)}) = \tau \cdot \prod_{d=1}^D \left[\theta + (1-\theta)N_X^{(d)} \right] \cdot \frac{N_X^{(d)}}{\theta + (1-\theta)N_X^{(d)}} = \tau \cdot \prod_{d=1}^D N_X^{(d)}, \quad (19)$$

which concludes the proof. \square

Proof for Proposition 2

Proof. According to the definition, the candidate patches sampled from $\text{SPP}(Y, \tau)$ (or $\text{SPP}(X, \tau)$) follows a homogeneous Poisson process with intensity S_Y (or S_X). Since there exists possibility to generate empty patches, we use intensity $S_X \cdot P(S_{\square^X} > 0)$ for thinning the Poisson process to generate nonempty patches. Given the same budget τ , Proposition 2 holds if we can prove the following equality of the two Poisson process intensities

$$S_Y \cdot P(S_{\pi_{Y,X}(\square^Y)} > 0) = S_X \cdot P(S_{\square^X} > 0). \quad (20)$$

Due to the independence of dimensions, $P(S_{\square^x} > 0)$ can be rewritten as

$$P(S_{\square^x} > 0) = \prod_d P(l_X^{(d)} > 0) \quad (21)$$

Assuming $N_X^{(d)} \geq 2$, we have

$$P(l_X^{(d)} > 0) = \frac{1}{N_X^{(d)}} + \frac{N_X^{(d)} - 1}{N_X^{(d)}} \cdot (1 - \theta) = \frac{1}{N_X^{(d)}} \cdot [\theta + N_X^{(d)}(1 - \theta)]. \quad (22)$$

W.l.o.g, we assume that the two arrays, X and Y , have the same shape apart from the d' th dimension where Y has one additional column (the general case of more columns follows by induction), then $S_X/S_Y = N_X^{(d')}/N_Y^{(d')}$.

There are two cases to consider: (1) X and Y share the terminal boundary in the d' th dimension; (2) X and Y share the initial boundary in the d' th dimension. In either case, by independence of dimensions, we have

$$P(S_{\pi_{Y,X}(\square^Y)} > 0) = P(\pi_{Y,X}(l_Y^{(d')}) > 0) \cdot \prod_{d \neq d'} P(l_Y^{(d)} > 0). \quad (23)$$

In case (1) where X and Y share the terminal boundary in the d' th dimension, we have

$$P(\pi_{Y,X}(l_Y^{(d')}) > 0) = \frac{\theta}{N_Y^{(d')}} + \frac{N_X^{(d')}}{N_Y^{(d')}} \cdot (1 - \theta) = \frac{1}{N_Y^{(d')}} \cdot [\theta + N_X^{(d')}(1 - \theta)], \quad (24)$$

where the first term $\frac{\theta}{N_Y^{(d')}}$ corresponds to the case that the initial position is on the boundary of Y and $l_Y^{(d')} \geq 2$ (otherwise \square^Y will not cross into X); while the second term $\frac{N_X^{(d')}}{N_Y^{(d')}} \cdot (1 - \theta)$ corresponds to the cases that the initial position is not on the boundary of Y . By combining Eqs. (21), (22), (23) and (24), we have

$$\frac{P(S_{\pi_{Y,X}(\square^Y)} > 0)}{P(S_{\square^x} > 0)} = \frac{P(\pi_{Y,X}(l_Y^{(d')}) > 0)}{P(l_X^{(d')} > 0)} = \frac{N_X^{(d')}}{N_Y^{(d')}} = \frac{S_X}{S_Y}. \quad (25)$$

Thus we have $S_Y \cdot P(S_{\pi_{Y,X}(\square^Y)} > 0) = S_X \cdot P(S_{\square^x} > 0)$.

In case (2) where X and Y share the initial boundary in the d' th dimension, we have

$$P(\pi_{Y,X}(l_Y^{(d')}) > 0) = \frac{1}{N_Y^{(d')}} + \frac{N_X^{(d')} - 1}{N_Y^{(d')}} \cdot (1 - \theta) = \frac{1}{N_Y^{(d')}} \cdot [\theta + N_X^{(d')}(1 - \theta)]. \quad (26)$$

The conclusion can be similarly derived. \square

Because of the same Poisson process intensity Eq. (20), the following equality also holds

$$P_{K_\tau, \{m_k\}_k}^Y \left(\pi_{Y,X}^{-1} \left(K_\tau^X, \{m_k^X\}_{k=1}^{K_\tau^X} \right) \right) = P_{K_\tau, \{m_k\}_k}^X \left(K_\tau^X, \{m_k^X\}_{k=1}^{K_\tau^X} \right) \quad (27)$$

Proof for Proposition 3

Proof. W.l.o.g, we assume that the two arrays, X and Y , have the same shape apart from the d' th dimension where Y has one additional column (the general case follows by induction). For dimensions $d \neq d'$, it is obvious that the law of patches are consistent under projection because the projection is the identity. Given the same budget τ , Proposition 3 holds if we can prove the following equality

$$P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) = P_u^X(u_X^{(d')} \mid |u_X^{(d')}| \geq 1), \quad (28)$$

where $u_X^{(d')}$ indicates the initial position, $s_X^{(d')}$, and the side-length, $l_X^{(d')}$, of the d' th side of \square^X ; $|u_X^{(d')}| \geq 1$ means that there is at least one “1” entry in $u_X^{(d')}$ as \square^X is a *nonempty* patch.

There are two cases to consider: (1) X and Y share the initial boundary in the d' th dimension; (2) X and Y share the terminal boundary in the d' th dimension. In each case, there are two cases (denoted as A & B in the following) regarding whether the terminal/initial (for Case 1/2, respectively) position locates at the boundary of X . In total we have four cases to discuss as follows.

In case (1) where X and Y share the initial boundary, $|\pi_{Y,X}(u_Y^{(d')})| \geq 1$ implies the condition of $s_Y^{(d')} \in X$ because $s_Y^{(d')} \notin X$ would lead to $|\pi_{Y,X}(u_Y^{(d')})| = 0$. Thus, the left term of Eq. (28) can be expressed as

$$\begin{aligned} P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) &= P_{s,l}^Y \left((s_Y^{(d')}, l_Y^{(d')}) \mid s_Y^{(d')} \in X \right) \\ &= P_s^Y \left(s_Y^{(d')} \mid s_Y^{(d')} \in X \right) P_l^Y \left(l_Y^{(d')} \mid s_Y^{(d')}, s_Y^{(d')} \in X \right). \end{aligned} \quad (29)$$

For convenience of notation, we let $\theta_{\dagger} = P_s^Y(s_Y^{(d')} \mid s_Y^{(d')} \in X)$, specifically $\theta_{\dagger} = \frac{1}{\theta + (1-\theta)N_X^{(d')}}$ if $s_Y^{(d')} = 1$; $\theta_{\dagger} = \frac{1-\theta}{\theta + (1-\theta)N_X^{(d')}}$ if $s_Y^{(d')} > 1$.

[Case 1.A] For $0 < s_X^{(d')} + l_X^{(d')} - 1 < N_X^{(d')} < N_Y^{(d')}$,

$$P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) = \theta_{\dagger} \theta^{l_X^{(d')} - 1} (1 - \theta) = P_u^X(u_X^{(d')} \mid |u_X^{(d')}| \geq 1). \quad (30)$$

[Case 1.B] For $0 < s_X^{(d')} + l_X^{(d')} - 1 = N_X^{(d')} < N_Y^{(d')}$,

$$\begin{aligned} &P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) \\ &= P_s^Y(s_Y^{(d')} \mid s_Y^{(d')} \in X) P_l^Y(l_Y^{(d')} = l_X^{(d')} \mid s_Y^{(d')}, s_Y^{(d')} \in X) \\ &\quad + P_s^Y(s_Y^{(d')} \mid s_Y^{(d')} \in X) P_l^Y(l_Y^{(d')} = l_X^{(d')} + 1 \mid s_Y^{(d')}, s_Y^{(d')} \in X) \\ &= \theta_{\dagger} \theta^{l_X^{(d')} - 1} (1 - \theta) + \theta_{\dagger} \theta^{l_X^{(d')}} = \theta_{\dagger} \theta^{l_X^{(d')} - 1} = P_u^X(u_X^{(d')} \mid |u_X^{(d')}| \geq 1) \end{aligned} \quad (31)$$

In case (2) where X and Y share the terminal boundary, $|\pi_{Y,X}(u_Y^{(d')})| \geq 1$ implies the condition of $s_Y^{(d')} \cdot l_Y^{(d')} > 1$ because $(s_Y^{(d')} = 1, l_Y^{(d')} = 1)$ would lead to $|\pi_{Y,X}(u_Y^{(d')})| = 0$. Thus, the left term of Eq. (28) can be expressed as

$$P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) = P_{s,l}^Y \left((s_Y^{(d')}, l_Y^{(d')}) \mid s_Y^{(d')} \cdot l_Y^{(d')} > 1 \right). \quad (32)$$

Given the condition of $s_Y^{(d')} \cdot l_Y^{(d')} > 1$ and the assumption $N_Y^{(d')} = N_X^{(d')} + 1$, we have

$$\begin{aligned} &P_s^Y \left(s_Y^{(d')} = 1 \mid s_Y^{(d')} \cdot l_Y^{(d')} > 1 \right) = \frac{P_{s,l}^Y(s_Y^{(d')} = 1, l_Y^{(d')} \geq 2)}{P_{s,l}^Y(s_Y^{(d')} \cdot l_Y^{(d')} > 1)} \\ &= \frac{1 \cdot \theta}{1 + (N_Y^{(d')} - 1)(1 - \theta) - 1 \cdot (1 - \theta)} = \frac{\theta}{\theta + (1 - \theta)N_X^{(d')}} \end{aligned} \quad (33)$$

and

$$\begin{aligned} &P_s^Y \left(s_Y^{(d')} = 2 \mid s_Y^{(d')} \cdot l_Y^{(d')} > 1 \right) = \frac{P_s^Y(s_Y^{(d')} = 2)}{P_{s,l}^Y(s_Y^{(d')} \cdot l_Y^{(d')} > 1)} \\ &= \frac{1 - \theta}{1 + (N_Y^{(d')} - 1)(1 - \theta) - 1 \cdot (1 - \theta)} = \frac{1 - \theta}{\theta + (1 - \theta)N_X^{(d')}}. \end{aligned} \quad (34)$$

[Case 2.A] For $\pi_{Y,X}(s_Y^{(d')}) = 1$, we have $s_X^{(d')} = 1$,

$$\begin{aligned}
& P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) = P_{s,l}^Y \left((s_Y^{(d')}, l_Y^{(d')}) \mid s_Y^{(d')} \cdot l_Y^{(d')} > 1 \right) \\
& = P_s^Y (s_Y^{(d')} = 2 \mid s_Y^{(d')} \cdot l_Y^{(d')} > 1) P_l^Y (l_Y^{(d')} = l_X^{(d')} \mid s_Y^{(d')} = 2) \\
& \quad + P_s^Y (s_Y^{(d')} = 1 \mid s_Y^{(d')} \cdot l_Y^{(d')} > 1) P_l^Y (l_Y^{(d')} = l_X^{(d')} + 1 \mid s_Y^{(d')} = 1) \\
& = \frac{1 - \theta}{\theta + (1 - \theta)N_X^{(d')}} \cdot \theta^{l_X^{(d')} - 1} \theta_{\ddagger} + \frac{\theta}{\theta + (1 - \theta)N_X^{(d')}} \cdot \theta^{l_X^{(d')} - 1} \theta_{\ddagger} \\
& = \frac{\theta^{l_X^{(d')} - 1} \theta_{\ddagger}}{\theta + (1 - \theta)N_X^{(d')}} = P_u^X (u_X^{(d')} \mid |u_X^{(d')}| \geq 1),
\end{aligned} \tag{35}$$

where $\theta_{\ddagger} = 1$ if $0 < \pi_{Y,X}(s_Y^{(d')}) + l_X^{(d')} - 1 = N_X^{(d')}$; $\theta_{\ddagger} = 1 - \theta$ if $0 < \pi_{Y,X}(s_Y^{(d')}) + l_X^{(d')} - 1 < N_X^{(d')}$.

[Case 2.B] For $\pi_{Y,X}(s_Y^{(d')}) > 1$, we have $s_X^{(d')} > 1$,

$$\begin{aligned}
& P_u^Y \left(\pi_{Y,X}^{-1}(u_X^{(d')}) \mid |\pi_{Y,X}(u_Y^{(d')})| \geq 1 \right) = \frac{1 - \theta}{\theta + (1 - \theta)N_X^{(d')}} \cdot \theta^{l_X^{(d')} - 1} \theta_{\ddagger} \\
& = P_u^X (u_X^{(d')} \mid |u_X^{(d')}| \geq 1).
\end{aligned} \tag{36}$$

Consider all D dimensions, for each case, we have $P_{\square}^Y (\pi_{Y,X}^{-1}(\square^X) \mid S_{\pi_{Y,X}(\square^Y)} > 0) = P_{\square}^X (\square^X \mid S_{\square^X} > 0)$. \square

Detail Proof of $P_{\boxplus}^Y (\pi_{Y,X}^{-1}(\boxplus_X)) = P_{\boxplus}^X (\boxplus_X)$

$$\begin{aligned}
& P_{\boxplus}^Y (\pi_{Y,X}^{-1}(\boxplus_X)) = P_{\boxplus}^Y \left(\pi_{Y,X}^{-1} \left(K_{\tau}^X, \{m_k^X, \square_k^X\}_{k=1}^{K_{\tau}^X} \right) \right) \\
& = P_{K_{\tau}, \{m_k\}_k}^Y \left(\pi_{Y,X}^{-1} \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \right) \\
& \quad \cdot P_{\square}^Y \left(\pi_{Y,X}^{-1} \left(\{\square_k^X\}_{k=1}^{K_{\tau}^X} \mid K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \mid S_{\pi_{Y,X}(\square^Y)} > 0 \right)
\end{aligned} \tag{37}$$

$$\begin{aligned}
& = P_{K_{\tau}, \{m_k\}_k}^Y \left(\pi_{Y,X}^{-1} \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \right) \\
& \quad \cdot P_{\square}^Y \left(\pi_{Y,X}^{-1} \left(\{\square_k^X\}_{k=1}^{K_{\tau}^X} \mid K_{\tau}^X \right) \mid S_{\pi_{Y,X}(\square^Y)} > 0 \right)
\end{aligned} \tag{38}$$

$$\begin{aligned}
& = P_{K_{\tau}, \{m_k\}_k}^Y \left(\pi_{Y,X}^{-1} \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \right) \\
& \quad \cdot \prod_{k=1}^{K_{\tau}^X} P_{\square}^Y \left(\pi_{Y,X}^{-1} (\square_k^X) \mid S_{\pi_{Y,X}(\square^Y)} > 0 \right)
\end{aligned} \tag{39}$$

$$\begin{aligned}
& = P_{K_{\tau}, \{m_k\}_k}^X \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \\
& \quad \cdot \prod_{k=1}^{K_{\tau}^X} P_{\square}^Y \left(\pi_{Y,X}^{-1} (\square_k^X) \mid S_{\pi_{Y,X}(\square^Y)} > 0 \right)
\end{aligned} \tag{40}$$

$$\begin{aligned}
& = P_{K_{\tau}, \{m_k\}_k}^X \left(K_{\tau}^X, \{m_k^X\}_{k=1}^{K_{\tau}^X} \right) \cdot \prod_{k=1}^{K_{\tau}^X} P_{\square}^X (\square_k^X \mid S_{\square^X} > 0) \\
& = P_{\boxplus}^X \left(K_{\tau}^X, \{m_k^X, \square_k^X\}_{k=1}^{K_{\tau}^X} \right) = P_{\boxplus}^X (\boxplus_X).
\end{aligned} \tag{41}$$

We can obtain Eq. (38) from Eq. (37) because

$$P \left(\{m_k, \square_k\}_{k=1}^{K_{\tau}} \mid K_{\tau} \right) = P \left(\{m_k\}_{k=1}^{K_{\tau}} \mid K_{\tau} \right) \cdot P \left(\{\square_k\}_{k=1}^{K_{\tau}} \mid K_{\tau} \right)$$

which indicates

$$P\left(\{\square_k\}_{k=1}^{K_\tau} | K_\tau, \{m_k\}_{k=1}^{K_\tau}\right) = P\left(\{\square_k\}_{k=1}^{K_\tau} | K_\tau\right)$$

We can obtain Eq. (39) from Eq. (38) because of independence of patches. Eq. (40) is derived from Eq. (39) by applying Proposition 2 and Eq. (41) is derived from Eq. (40) by applying Proposition 3.

Multiple-Try Metropolis for Sampling $\{r_i\}_i, \{c_j\}_j$ (in Section 5.2)

The prior distributions of $\{r_i\}_i$ and $\{c_j\}_j$ are discrete uniform distributions over all the $N!$ permutations. To cooperate with the C-SMC sampler for higher acceptance ratio, we adopt the Multiple-Try Metropolis method [16] for sampling the row and column indices of the relational data. For each r_i , we propose an exchange between r_i and Z proposal rows $\{r_{i'_z}\}_{z=1}^Z$, which are randomly chosen. The detail for sampling $\{r_i\}_i$ is summarized in Algorithm 3 (sampling $\{c_j\}_j$ is similar).

Algorithm 3 Multiple-Try Metropolis for Sampling $\{r_i\}_i$

Input: Relational data R , number of proposals Z , $\{\rho_{ij}\}_{i,j}$, $\{r_i\}_i$ and $\{c_j\}_j$

Output: New assignments of $\{r_i\}_i$

for $i = 1, \dots, N$ **do**

Propose Z independent proposal indices $\{r_{i'_z}\}_{z=1}^Z$ for exchanging with r_i ;

Compute the weights $\{\omega(r_{i'_z})\}_{z=1}^Z$ as

$$\omega(r_{i'_z}) = \frac{\prod_j \ell(r_{i'_z}, c_j, \rho_{ij}) \ell(r_i, c_j, \rho_{i'_z j})}{\prod_j \ell(r_i, c_j, \rho_{ij}) \ell(r_{i'_z}, c_j, \rho_{i'_z j})}; \quad (42)$$

Sample r_{i^*} from $\{r_{i'_z}\}_{z=1}^Z$ with probability in proportional to $\{\omega(r_{i'_z})\}_{z=1}^Z$;

Suppose r_i and r_{i^*} are exchanged; propose $Z - 1$ new independent proposal indices $\{r_{i''_z}\}_{z=1}^{Z-1}$ and set $r_{i''_Z} = r_i$ for exchanging with r_{i^*} ;

Compute the weights $\{\omega(r_{i''_z})\}_{z=1}^Z$ as

$$\omega(r_{i''_z}) = \frac{\prod_j \ell(r_{i''_z}, c_j, \rho_{ij}) \ell(r_{i^*}, c_j, \rho_{i''_z j})}{\prod_j \ell(r_{i^*}, c_j, \rho_{ij}) \ell(r_{i''_z}, c_j, \rho_{i''_z j})} \quad (43)$$

Accept the exchange between r_i and r_{i^*} with the ratio

$$\alpha = \min\left(1, \frac{\omega(r_{i'_1}) + \dots + \omega(r_{i'_Z})}{\omega(r_{i''_1}) + \dots + \omega(r_{i''_Z})}\right) \quad (44)$$

end for
